REMARKS

I. General

Claims 1-18 were pending in the present application, and all of the pending claims are rejected in the current Office Action (mailed March 4, 2005). The outstanding issues raised in the current Office Action are:

- Claims 1-18 are rejected under 35 U.S.C. § 103(a) as being unpatentable over "The HTTP Distribution and Replication Protocol," http://www.w3.org/TR/NOTE, August 25, 1997 (hereinafter "*DRP*") in view of U.S. Patent No. 5,734,898 issued to He (hereinafter "*He*").

In response, Applicant respectfully traverses the outstanding claim rejections, and requests reconsideration and withdrawal thereof in light of the amendments and remarks presented herein.

II. New Claim(s)

New claims 19-20 are added herein. Claim 19 depends from claim 1 and recites that the informing comprises "responsive to determining said digest value has a match in said digest index, performing said informing." Claim 20 depends from claim 10 and recites that the logic code for informing comprises "logic code for performing said informing responsive to said logic code for comparing determining that said received digest value has a match in said digest index." No new matter is added by these claims. Further such claims are believed to be allowable at least because of their dependency from their respective independent claims for the reasons asserted further below for claims 1 and 10.

III. Rejections Under 35 U.S.C. § 103(a)

To establish a prima facie case of obviousness, three basic criteria must be met. *See* M.P.E.P. § 2143. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the applied references must teach or suggest all the claim limitations. Without conceding any other criteria, Applicant respectfully asserts that the rejection does not satisfy at least the third criteria because the combination of *DRP* and *He* fails to teach or suggest all elements of the claims, as discussed further below.

Independent Claim 1

The combination of *DRP* and *He* fails to teach or suggest all elements of independent claim 1. For instance, independent claim 1 recites:

> A method for reducing network retrieval latency, comprising the steps of:
> sending a request for a data object to a server;
> receiving a header portion of a response to said request;
> parsing said header portion for a digest value;
> comparing said digest value to a digest index;
> retrieving a cached data object from a cache if said digest value has a match in said digest index;
> sending said cached data object to a client; and
> if said digest value has a match in said digest index, informing said server to stop sending a remaining portion of said response. (Emphasis added).

The combination of *DRP* and *He* fails to teach or suggest all of the above elements of claim 1. The current Office Action relies on *DRP* as teaching all of the elements except for "informing said server to stop sending a remaining portion of said response", which the Office Action relies on *He* as teaching. However, *DRP* does not teach parsing a received header portion of a response for a digest value, comparing the digest value to a digest index, and retrieving a cached data object from a cache if the digest value has a match in the digest index. A brief overview of *DRP* is provided below.

In *DRP*, a client requests an index for a site, and the index includes content identifiers for the files of the site. Thus, the index provides a hierarchical snapshot of the set of files for a site at a particular moment in time, *see* Section 2.2 of *DRP*. The client can then download all of the files of the index. The client can further use the content identifiers to determine if any of the files in an index have the same content, and avoid downloading duplicative content. A new index can later be requested to determine if an update/modification is available for the content. That is, the client can request the index from the site and compare the content identifiers included in the newly received index against the content identifiers of the old index to determine if any of the content has changed. If the content has changed, the client can request the updated content using a request with a Content-ID or Differential-ID included in the header of the request.

In *DRP*, a comparison of two digest values can occur in any of the following contexts:

(1) a client retrieves an index file (with digests for many files) from a server: the client extracts a digest from the received index, and the client compares against the digest for client-cache entry;

(2) the client sends the digest (as Content-ID) to the server: the server compares the digest against its available content, and returns to the client content, if any, that matches; and

(3) client sends the digest (as Content-ID) via a proxy: the proxy compares the digest against cached content, and on a match returns the cache entry; otherwise, the proxy forwards the digest to the server as in context (2).

In no case of *DRP* does a server send, in response to a client request, a digest for the current request, and the digest comparison is then made at the client or proxy cache. In contexts (2) and (3), *DRP* makes the digest comparison at the recipient of a REQUEST, rather than making the comparison at the recipient of a RESPONSE to a request. In context (1), *DRP* sends an index file full of digest values, on the speculation that the client might want to retrieve content corresponding to one or more of those digest values. Such index file is not sent in response to a client request for content.

DRP does not teach or suggest a client requesting an object, receiving a response to that request where the response includes a header having a digest value, and comparing the digest value against an index to determine if the requested object is in the cache. Rather, in *DRP* the client requests a separate index file that includes the content identifiers. *DRP* also describes using a content-ID or differential-ID in a header. However, *DRP* does not teach that the client uses these IDs for comparing against an index to determine if the requested object is in the cache. Claim 1 recites receiving a digest value in the header portion of a response to a request for a data object. Thus, embodiments of the present invention could be employed to alleviate requesting an index file by the client in the manner required by *DRP*.

With regard specifically to claim 1, *DRP* does not teach or suggest all elements of claim 1 that the present Office Action relies on *DRP* as teaching. First, *DRP* does not teach or suggest that the client compares a digest value (or the content-ID) received in a response header against an index and retrieves a cached data object from a cache if the digest value has a match in the index. Rather, using the *DRP* scheme, the client would check the content-ID against the index for a site to determine if the desired content is already possessed by the client. If the content-ID matches that of the corresponding file in the index, then the client could forego downloading this content again. In this case, the content-ID that is compared against the index for a site is not included in a header of a response to a request for an object, as no request for the object is made to the server.

Further, as mentioned above, a content-ID and a differential-ID may be used as header fields in *DRP*. However, in the case in which these IDs are used, the content-ID or Differential-ID is not compared against the index, and thus a cached data object is not retrieved from the cache if a match occurs (as no comparison is made). Rather, in the instances in which these IDs are included in a header in *DRP*, the information identified by such IDs is to be retrieved from a server. Any comparison that is made in this instance, is made at the server to determine the corresponding content to return to the client.

Second, claim 1 recites "informing said server to stop sending a remaining portion of said response." *DRP* does not inform the server to stop sending a remaining portion of the response. In *DRP*, a client retrieves an index file (with digests for many files) from a server: the client extracts a digest from the received index, and the client compares against the digest for client-cache entry. If the digest contained in the index file matches the digest for a cache entry, the client may forego requesting the data from the server. However, in the instance in which the client's cache is to be updated, the client may request the updated data from the server. In making such a request, the client may send the digest (as Content-ID) to the server, wherein the server compares the digest against its available content and returns to the client content, if any, that matches. Thus, in the case in *DRP* in which a header includes a Content-ID, the server does not stop sending a remaining portion of the response, as the full response is needed for updating the client's content.

The Office Action appears to concede that *DRP* does not teach "informing said server to stop sending a remaining portion of said response." However, the Office Action asserts, on page 3 thereof, that this teaching is arguably inherent in the teaching of the applied *DRP* reference. However, to be inherent, the element must <u>necessarily flow</u> from the teaching of the *DRP* reference. The Office Action has failed to explain how this element necessarily flows from the teaching of the *DRP* reference, and thus has failed to establish inherency. Further, such teaching does not necessarily flow from the teaching of the *DRP* reference. As described above, in the instance in which a digest contained in a received index file matches the digest of a client's cache entry, a request for such data need not be made to the server. Further, in any instance in which a Client-ID is contained in a header in *DRP*, the corresponding response from the server is needed by the client in order to update its cache, and thus it would not necessarily flow to inform the server to stop sending any remaining portion of such a response (as the client desires the full response to be assured that its cache data is fully updated).

Further, the Office Action asserts that *He* teaches this element, and concludes that it would have been obvious to one of ordinary skill in the art at the time of the invention to modify *DRP* to include this teaching of *He*. However, *He* does not teach or suggest "informing said server to stop sending a remaining portion of said response." Rather, the portion of *He* relied upon by the Office Action refers to aborting a "transaction" (i.e., as opposed to committing the transaction), and does not refer to informing a server to stop sending a remaining portion of a response. More particularly, the Office Action cites to col. 3, line 32-40 of *He*, which provides:

> FIG. 20 shows the operation of client A requesting a commit by the server of the update request described above. Client A sends commitRequest() to the server. Then the server returns commitReturn(). Similarly, for abort operation, an abort request and abort return are transferred. In this case, since the server version and client A version of object oid1 have already been updated to the same, the server will not send a version and other data to client A. This is a waste in terms of the usage of communication line.

This portion of *He* is addressing the performance of transactions and maintaining data objects consistent between a client and a server for performance of a transaction. Performing actions as "transactions" is well-known in the art, and one of ordinary skill in the art would readily appreciate that the above teaching of *He* is addressing aborting a transaction, as opposed to committing the transaction. Aborting the transaction is not referring to informing the server to stop sending a remaining portion of a response. The above portion of *He* also mentions that the server will not send a version and other data to client A because the versions of object oid1 contained on the server and client already match. Thus, this refers to the server foregoing the sending of data to the client, rather than the client informing the server to stop sending a "remaining portion" of a response. Thus, each of *DRP* and *He* fails to teach or suggest this element of claim 1, and as such the applied combination of *DRP* and *He* does not render claim 1 obvious under 35 U.S.C. §103(a).

Independent Claim 6

Independent claim 6 recites:

> A method for reducing network retrieval latency, comprising the steps of:
>
> sending a request for a data object to a server;
> receiving a server response from said server;
> calculating a digest value for said data object based on said server response;
> sending a response to a client cache starting with a header portion, said header portion including said digest value and enabling said client cache to compare said digest value to a digest index, retrieve a cached data object from said client cache if said digest value has a match in said digest index, and send said cached data object to a client; and
> upon receiving a message from said client cache to stop sending said response, stopping the sending of said response. (Emphasis added).

The combination of *DRP* and *He* fails to teach or suggest all of the above elements of claim 6. The current Office Action relies on *DRP* as teaching all of the elements except for "upon receiving a message from said client cache to stop sending said response, stopping the sending of said response", which the Office Action relies on *He* as teaching. However, *DRP* does not teach "sending a response to a client cache starting with a header portion, said header portion including said digest value and enabling said client cache to compare said digest value to a digest index, retrieve a cached data object from said client cache if said digest value has a match in said digest index, and send said cached data object to a client" (emphasis added).

As discussed above with claim 1, in *DRP* a server sends the client an index file containing digest values for various files available at a site provided by the server. *DRP* does not teach or suggest including a digest value in a header portion of a response to a client.

Further, neither *DRP* nor *He* teaches or suggests "upon receiving a message from said client cache to stop sending said response, stopping the sending of said response". As discussed above with claim 1, *DRP* neither expressly nor inherently teaches receiving from a client a message to stop sending a response. As also discussed above with claim 1, *He* does not teach or suggest this element either.

In view of the above, claim 6 is not obvious under 35 U.S.C. §103(a) because the applied combination of *DRP* and *He* fails to teach or suggest all elements of claim 6.

<u>**Independent Claim 7**</u>

Independent claim 7 recites:

> A method for reducing network retrieval latency, comprising the steps of:
> receiving a first request for a data object;
> obtaining a digest value of said requested data object;
> <u>inserting said digest value into a header portion of a response;</u>
> <u>sending said response, starting with said header portion</u>; and
> <u>upon receiving a second request to stop sending said response, stopping the sending of said response</u>. (Emphasis added).

The combination of *DRP* and *He* fails to teach or suggest all of the above elements of claim 7. The current Office Action relies on *DRP* as teaching all of the elements except for "upon receiving a second request to stop sending said response, stopping the sending of said response", which the Office Action relies on *He* as teaching. However, *DRP* does not teach "inserting said digest value into a header portion of a response" and "sending said response, starting with said header portion".

As discussed above with claim 1, in *DRP* a server sends the client an index file containing digest values for various files available at a site provided by the server. *DRP* does not teach or suggest inserting a digest value into a header portion of a response to a client.

Further, neither *DRP* nor *He* teaches or suggests "upon receiving a second request to stop sending said response, stopping the sending of said response". As discussed above with claim 1, *DRP* neither expressly nor inherently teaches receiving from a client a request to stop sending a response. As also discussed above with claim 1, *He* does not teach or suggest this element either.

In view of the above, claim 7 is not obvious under 35 U.S.C. §103(a) because the applied combination of *DRP* and *He* fails to teach or suggest all elements of claim 7.

Independent Claim 10

Independent claim 10 recites:

> A computer program product for use in conjunction with a computer system for reducing network retrieval latency, comprising:
> logic code for sending a request for a data object to a server;
> logic code for receiving a header portion of a response to said request;
> logic code for parsing said header portion for a digest value;
> logic code for comparing said digest value to a digest index;
> logic code for retrieving a cached data object from a cache if said digest value has a match in said digest index;
> logic code for sending said cached data object to a client; and
> logic code for informing said server to stop sending a remaining portion of said response.

The combination of *DRP* and *He* fails to teach or suggest all of the above elements of claim 10. The current Office Action relies on *DRP* as teaching all of the elements except for "logic code for informing said server to stop sending a remaining portion of said response", which the Office Action relies on *He* as teaching. However, *DRP* does not teach "logic code for parsing said header portion for a digest value" and "logic code for retrieving a cached data object from a cache if said digest value has a match in said digest index".

As discussed above with claim 1, in *DRP* a server sends the client an index file containing digest values for various files available at a site provided by the server. *DRP* does not teach or suggest including a digest value in a header portion of a response to a client. Thus, *DRP* does not teach or suggest logic code for parsing a header portion of a response received by a client for a digest value, as the digest values are contained in a separate index file rather than in the header portion of a response in *DRP*.

Further, neither *DRP* nor *He* teaches or suggests "logic code for informing said server to stop sending a remaining portion of said response". As discussed above with claim 1, *DRP* neither expressly nor inherently teaches a client informing said server to stop sending a remaining portion of a response. As also discussed above with claim 1, *He* does not teach or suggest this element either.

In view of the above, claim 10 is not obvious under 35 U.S.C. §103(a) because the applied combination of *DRP* and *He* fails to teach or suggest all elements of claim 10.

Independent Claim 15

Independent claim 15 recites:

> A computer program product for reducing network retrieval latency, comprising:
> logic code for sending a request for a data object to a server;
> logic code for receiving a server response from said server;
> logic code for calculating a digest value for said data object based on said server response;
> logic code for sending a response to a client cache starting with a header portion, said header portion including said digest value and enabling said client cache to compare said digest value to a digest index, retrieve a cached data object from said client cache if said digest value has a match in said digest index, and send said cached data object to a client; and
> logic code for stopping the send of said response upon receiving a message from said client cache to stop sending said response. (Emphasis added).

The combination of *DRP* and *He* fails to teach or suggest all of the above elements of claim 15. The current Office Action relies on *DRP* as teaching all of the elements except for "logic code for stopping the send of said response upon receiving a message from said client cache to stop sending said response", which the Office Action relies on *He* as teaching. However, *DRP* does not teach "logic code for sending a response to a client cache starting with a header portion, said header portion including said digest value and enabling said client cache to compare said digest value to a digest index, retrieve a cached data object from said client cache if said digest value has a match in said digest index, and send said cached data object to a client" (emphasis added).

As discussed above with claim 1, in *DRP* a server sends the client an index file containing digest values for various files available at a site provided by the server. *DRP* does not teach or suggest including a digest value in a header portion of a response to a client.

Further, neither *DRP* nor *He* teaches or suggests "logic code for stopping the send of said response upon receiving a message from said client cache to stop sending said response". As discussed above with claim 1, *DRP* neither expressly nor inherently teaches receiving from a client a message to stop sending a response. As also discussed above with claim 1, *He* does not teach or suggest this element either.

In view of the above, claim 15 is not obvious under 35 U.S.C. §103(a) because the applied combination of *DRP* and *He* fails to teach or suggest all elements of claim 15.

- **Independent Claim 16**

Independent claim 16 recites:

> A computer program product for reducing network retrieval latency, comprising:
> logic code for receiving a first request for a data object;
> logic code for obtaining a digest value of said requested data object;
> logic code for inserting said digest value into a header portion of a response;
> logic code for sending said response, starting with said header portion; and
> logic code for stopping the sending of said response upon receiving a second request to stop sending said response. (Emphasis added).

The combination of *DRP* and *He* fails to teach or suggest all of the above elements of claim 16. The current Office Action relies on *DRP* as teaching all of the elements except for "logic code for stopping the sending of said response upon receiving a second request to stop sending said response", which the Office Action relies on *He* as teaching. However, *DRP* does not teach "logic code for inserting said digest value into a header portion of a response" and "logic code for sending said response, starting with said header portion".

As discussed above with claim 1, in *DRP* a server sends the client an index file containing digest values for various files available at a site provided by the server. *DRP* does not teach or suggest inserting a digest value into a header portion of a response to a client.

Further, neither *DRP* nor *He* teaches or suggests "logic code for stopping the sending of said response upon receiving a second request to stop sending said response". As discussed above with claim 1, *DRP* neither expressly nor inherently teaches receiving from a client a request to stop sending a response. As also discussed above with claim 1, *He* does not teach or suggest this element either.

In view of the above, claim 16 is not obvious under 35 U.S.C. §103(a) because the applied combination of *DRP* and *He* fails to teach or suggest all elements of claim 16.

Dependent Claims

Claims 2-5, 8-9, 11-14, and 17-18, which each depend directly and indirectly from one of independent claims 1, 7, 10, and 16 and thus inherit the limitations of their respective independent claims, are asserted as patentable over the 35 U.S.C. § 103 rejections at least for the reasons stated above for their respective independent claims.

IV. Conclusion

In view of the above, Applicant believes the pending application is in condition for allowance.

Applicant believes no fee is due with this response. However, if a fee is due, please charge our Deposit Account No. 08-2025, under Order No. 200308263-1 from which the undersigned is authorized to draw.

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail, Label No. EV 482744233US in an envelope addressed to: M/S Amendment, Commissioner for Patents, Alexandria, VA 22313.

Date of Deposit: June 6, 2005

Typed Name: Gail L. Miller

Signature: Gail L. Miller

Respectfully submitted,

By:

Jody C. Bishop
Attorney/Agent for Applicant(s)
Reg. No. 44,034
Date: June 6, 2005
Telephone No. (214) 855-8007